Computer Systems Validation

Orlando Lopez
Johnson & Johnson, NCS, Raritan, New Jersey, U.S.A.

INTRODUCTION

Computer systems validation, as established in *21 Code of Federal Regulation* (CFR) Part 11.10(a) and defined in the recent draft United States (US) Food and Drug Administration (FDA) guideline,[1] is one of the most important requirements applicable to computer systems performing FDA-regulated operations. It involves establishing the conformance to the intended use, user, regulatory, safety, and function allocated to the computer system.

Similar to any FDA-regulated products, quality is built into a computer system during its conceptualization, development, and operational life. The quality of computer systems cannot be tested after being developed. In addition to software and hardware testing, other verification activities include code walkthroughs, dynamic analysis, and trace analysis. The documentation generated during the validation can be subject to examination by FDA field investigators. The results of a high-quality validation program can ensure with a high degree of assurance the trustworthiness of electronic records and computer systems-related functionality.

OVERVIEW

The introduction in 1997 of 21 CFR Part 11, Electronic Records, Electronic Signatures Rule (hereafter referred to as Part 11) provided the formal codification applicable to computer systems performing FDA-regulated operations. One fundamental principle in Part 11 is that it requires organizations to store regulated electronic data in their electronic form once a record is saved to durable media, rather than keep paper-based printouts of the data on file, as had been the long-term practice in organizations performing regulated operations.[2] If information is not recorded to durable media, the stored data will be lost and they cannot be retrieved for future use. If ''retrievability'' is an attribute, then procedural and technological controls contained in Part 11 are essential to ensuring integrity. The implementation of procedural and technological controls to achieve compliance with Part 11 shall be monitored during the SLC.

The approach to be used in covering computer systems validation is by presenting key elements applicable to any development/maintenance methodology. It is not intended to cover everything that computer system validation should encompass, including Part 11. A wide range of information about computer systems validation is listed in the bibliography.

KEY VALIDATION ELEMENTS

The key elements required to successfully execute computer system validation projects are as follows:

Selection of a development/maintenance methodology that best suits the nature of the system under development.
Identification of operational functions associated with the users, operational checks, regulatory, company standards, and safety requirements.
Selection of hardware based on capacity and functionality.
Inspection and testing of the operational functions.
Identification and testing of ''worst case'' operational/production conditions.
Reproducibility of the testing results based on statistics.
Documentation of the validation process.[3]
Written design specification that describes what the software is intended to do and how it is intended to do it.
A written validation plan based on the design specification, including both structural and functional analysis.
Test results and evaluation of how these results demonstrate that the predetermined design specification has been met.
Availability of procedural controls to maintain the validation state of the computer system and its operating environment.
Any modification to a component of the system and its operating environment must be evaluated to determine the impact to the system. If required, qualification/validation is to be re-executed totally or partially.

Selection of a Development/Maintenance Methodology

The SLC is the ''period of time that begins when a product is conceived and ends when the product is no longer available for use.''[4] Certain overall discrete work

Encyclopedia of Pharmaceutical Technology
DOI: 10.1081/E-EPT 120018214

MARCEL DEKKER, INC.
270 Madison Avenue, New York, New York 10016

products are expected when evidencing the development and maintenance work of computer systems compliance to regulatory requirements. Refer to ''Documentation of the Validation Process.'' The selected SLC specifies the overall periods and associated events. Different system acquisition strategies and software development models can be adapted to the SLC depicted in Fig. 1.[5] The SLC model focuses on software engineering key practices and does not specify or discourage the use of any particular software development method. The acquirer determines which of the activities outlined by the standard will be conducted, and the developer is responsible for selecting the methods that support the achievement of contract requirements. A modifiable framework must be tailored to the unique characteristics of each project. The SLC includes the following periods:

Conceptualization.
Development.
Early operational life.
Maturity.
Aging.

Project Recommendation, Project Initiation, Release for Use, and Retirement are events. These events are considered phase gates or major decision points, which include formal approvals before the development can proceed to the next period.

The development methodology associated with the SLC is a structured process that decomposes the engineering tasks and associated work products in support of the computer system validation effort. It breaks down the systems development process into subperiods, containing specific inspection and testing tasks that are appropriate for the intended use of the computer system. During each subperiod, detailed discrete work products are developed. This approach leads to well-documented systems that are easier to test and maintain, and for which an organization can have confidence that the system's functions will be fulfilled with a minimum of unforeseen problems.

The most common development methodologies are the Waterfall Model, Incremental Development, Evolutionary Model, Object Oriented, and Spiral Model.

A critical component of the validation process is providing assurance that the development/maintenance methodology is being followed. The SLC and associated development/maintenance methodology applicable to computer systems performing regulated operations shall be specified in procedural control(s). A project team should have the authority to select a developmental/maintenance methodology that best suits the nature of the system under development/maintenance and that is different from the one included in the related procedural control. If this is the case, the selected development or maintenance methodology must be explained in the validation plan.

It is the objective of FDA-regulated companies to select the appropriate SLC and associated development/maintenance methodology. Development and mainten-

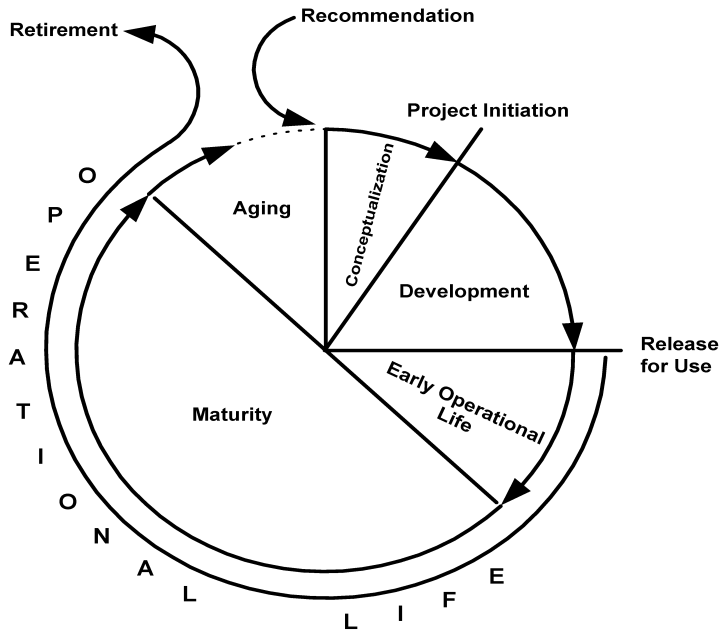

Fig. 1

Marcel Dekker, Inc.
270 Madison Avenue, New York, New York 10016

ance teams shall receive adequate training in the use of the chosen methodology.

Identification of Operational Functions

Using a structured process, the goal of the Development Period is to specify, design, build, test, and install the application to be automated or updated. One of the deliverables of the Development Period is the written and approved operational functions (e.g., user's, functional, design) that describe what the application is intended to do.

A key activity to identify operational functions is by gathering systems requirements. The term ''requirement'' defines a bounded characterization of the scope of the system. It contains the information essential to support the operation/operators. These requirements include functional capacity, execution capability, safety, operational, installation, system maintenance, and regulatory compliance.

The refined scope is captured in the requirements specification, which describes what the system is supposed to do from the process/user's/compliance perspective. The requirements specification is used as part of the framework to select the computer technology supplier and/or contract developer. The system functionality must be well defined at the outset in order to provide the prospective supplier/integrator with enough information to provide a detailed and meaningful quotation. The requirements specification is used to develop the performance qualification (PQ) protocol.

The requirements specification addresses the following:

The process to familiarize the developer with the user, process and data acquisition requirements, and special considerations of the project.
The scope of the system and strategic objectives.
The problem to be solved.
Process review and sequencing, as well as where each operation is to be completed.
The direction to solve the problem (e.g., device driven or may just be the mode of presentation of data, data security, data backup, data and status reporting and trending).
Redundancy and error-detection protocol.
Environmental control.
Interfaces (e.g., to field devices, data acquisition, reports and HMI), input/output (I/O) list, communication protocol, and data link requirements.
Type of control/process to be performed.
Operational checks and sequencing.
Data management.
Definition of the input and output domains.
How the data are to be collected, used, and stored.
How input data influence the operation of the system.
Retention requirements.
Data security requirements.
Audit trails and metadata.
Timing requirements.
Regulatory requirements.
Preliminary evaluation of technology.
Feasibility study and preliminary risk assessment.
Safety and security considerations.
Nonfunctional requirements (e.g., development standards, program-naming convention standards).

Each requirement in the requirements specification must be ''testable.'' A ''testable'' requirement includes an objective criterion and it is nonambiguous. A ''testable'' requirement provides the advantage that it can be recorded in quantified terms and allows for a subsequent review and independent evaluation of the test results.

Selection of Hardware Based on Capacity and Functionality

Based on the identification of operational functions and design, computer hardware technologies can be selected. Depending on available technology and cost, automated functions can be assign to the computer hardware or software. Computer hardware can be further decomposed into a number of subelements. Processor, memory, I/O and networks are some examples.

Process, field instruments, control requirements, available technology and cost are some of the factors driving the selection of the hardware. This selection is specified in the requirement specification and the implementation described in the design specification. The design specification needs to be sufficiently detailed in order to familiarize the implementation team (e.g. engineering) and the hardware vendor with the requirements and special considerations of the process, field instrument, and control requirements.

The requirements include, but are not limited to, the following:

Purpose of the system.
Regulatory requirements.
Information and material inputs.
Preliminary block diagrams.
Data processing requirements (e.g., supervisory control and data acquisition).
Number and type of I/O cards.
Instrumentation and cabling.
Control and information outputs.

Operating modes.
Alarms and alerts.
Safety features.
Error checking.
Reporting.
Redundancy requirements.
Environmental requirements.
Network requirements.
Supporting utilities requirements.
Hardware/human machine interfaces.
General plan and acceptance criteria.

The critical field instrumentation must support accuracy and reliability requirements over the entire process range conditions.

Inspection and Testing of the Operational Functions

Software engineering practices may include documented unit testing, code reviews, explicit high-level and low-level design documents, explicit requirements and functional specifications, structure charts and data flow diagrams, function-point analysis, defect and resolution tracking, configuration management, and a documented software development process.

These are the same quality principles that the FDA expects to be used during the development and maintenance of computer systems. These quality principles shall be contained in procedural controls. Inspections and testing are part of these principles.

Testing and inspections are activities performed as part of the development methodology. Numerous inspection steps are undertaken throughout the system development and operational life to determine whether a computer system is validated. These include static analyses such as document and code inspections, walk-through, and technical reviews. These activities and their outcomes help to reduce the amount of system-level functional testing needed in the operational environment in order to confirm that the software meets the requirements and intended uses.

Reproducibility of the Testing Results Based on Statistics

Testing is not just executing a program using a test data file or randomly selected test cases just prior to implementation. It is an on going process using techniques based on Statistical Process Control (SPC) principles and product quality concepts implemented as components of a Statistical Quality Control (SQC) program.[6]

If software systems are viewed as a manufacturing facility that produces the desired output products, then statistical sampling procedures and statistical inference can be used to predict the reliability of test results. Instead of paying too much attention to the development of the application, another factor that requires attention is the data (raw material) to be converted into information (product). Information flow is a design technique that may be helpful in achieving this task.

Concerning testing, the input and output domains must be strictly defined. This suggests a response to the following: 1) What sampling technique will ensure an adequate subset of possible input values that will provide as complete a test of the software as possible? and 2) What information sampling procedures will allow the developers to determine product reliability?

It is suggested that White and Black box test cases design strategies be used to sample the data (input domain), including Equivalent Partitioning, Boundary Analysis and Error Guessing, Cause–Effect Graphing, and Structural Tableau.

The basics of Software Testing must be understood before the more abstract principles of statistical inference are tackled.

Documentation of the Validation Process

Design specification

System requirements are allocated to the software design. During the technical design it is described how each specification described in the system specification deliverable is to be implemented. This includes also developed subsystems components and interfaces, data structure, design constraints, algorithms and system decomposition. This activity is very critical to medical device companies. Design inputs are contained in the requirements of the computer system, and design output(s) can be included as part of the specification of the design.[7] This design is the input for developing integration test and operational checks.

Design according to the development methodology and specific procedural controls.

Computer hardware and software architecture.
Data structures.
Flow of information.
Interfaces
Put together the design.
Perform design reviews. Verify whether the risks previously identified were mitigated as part of the solution presented in the design.
Finalized the test planning.
Design Part 11 technical controls.

Marcel Dekker, Inc.
270 Madison Avenue, New York, New York 10016

Approve the design specification deliverable.
Conduct in-process audit activities associated wit the technical design.
Re-visit the risk analysis.

Begin the planning of development of procedural controls for those Part 11 requirements not covered by technology.

Validation plan

Validation plans are documents that tailor a firm's overall philosophies, intentions, and approaches to be used for establishing performance adequacy to a specific project. They state who is responsible for performing development and validation activities. They identify which systems are subject to validation, define the nature and extent of inspection and testing expected to be done on each system, and outline the protocols to be followed to accomplish the validation.

In summary, validation plans describe the following:

Organizational structure of the computerization project.
Responsible departments and/or individuals.
Resource availability.
Risk management.
Time restrictions.
SLC and development methodology to be followed.
Deliverable items.
Overall acceptance criteria.
Development schedule and timeline.
System release sign-off process.
Sample format for key documentation.

Test results and evaluation

Although installation qualification (IQ)/operational qualification (OQ)/PQ terminology has served its purpose well and is one of the many legitimate ways to organize computer system testing tasks in FDA-regulated industries, this terminology may not be well understood among many software professionals. However, organizations performing regulated operations must be aware of these differences in terminology as they ask for and provide information regarding computer systems.

Once the qualification protocols have been completed, test results and data need to be formally evaluated. Written evaluation needs to be presented clearly in a manner that can be readily understood. The report should also address any nonconformance or deviation to the validation plan encountered during the qualification and resolution. The outline of the report parallels the structure of the associated protocol. The qualification testing should be linked with relevant specification's acceptance criteria, such as PQ vs. system requirements specification deliverable, OQ vs. system specification deliverable, and IQ vs. technical design specification deliverable. If applicable, it is included as part of the summary of the results of inspections and technical review of all technologies that are elements of the systems.

In very large validation efforts, a report references (by title and document reference number) other documents that satisfy the protocol requirements. In smaller validation efforts, actual evidence is incorporated as appendices to the report.

The documentation and results of the qualification efforts are assembled and reviewed by appropriate and qualified personnel. Following the review, the personnel responsible for the criticality of the system, including QA, approve the qualification effort.

The approval of all qualification reports is a confirmation that the computer system as a whole has been proven to fit its purpose and that all essential elements of documentation are available. On computer systems controlling manufacturing equipment (process control systems), the approval of all qualification reports indicates the release of the computerized systems to the Process/Product Performance Qualification. On other computer systems, the approval of all reports indicates the release of the system to the user.

Test results on Part 11 shall be addressed in the associated qualification report.

The Project Report summarizes the outcome of each activity performed to develop or maintain computer systems and the verification of critical checkpoints throughout the entire development process. The end-result is to verify that good quality development procedures were adhered to as established in the project plan.

All verification and testing results completed during the project shall be addressed in the Project Report as well.

The approval of the project report is the event to be considered prior to the release of the system for operation.

Validation maintenance

After the system has been released for operation, computer system maintenance activities take over. The maintenance activities must be governed by the same procedures followed during the Development Period.

The validated status of computer systems performing regulated operations is subject to threat of changes in its operating environment, either known or unknown. Adherence to security, operational management, business continuity, change management, periodic review, and decommissioning provides a high degree of assurance that the system is being maintained in a validated state. It is

the objective of organizations to have procedures in place to minimize the risk of computer systems performing regulated operations out of validated state.

Maintenance in computer systems becomes an essential issue, particularly when a new version of the supplier-provided standard software is updated. A change control procedure must be implemented whereby changes in the software and computer hardware may be evaluated, approved, and installed.

If necessary, additional analysis may be needed to evaluate the changes (e.g., impact analysis) to the computer systems. The procedure should allow for both planned and emergency changes to the system. This procedure must include provision for updating of pertinent documentation on the system, including procedures. Records of changes to the system must be kept for the same period as any other regular production document.

Table 1[8] summarizes the periods and events applicable to the operational life of computer systems and associated key practices.

Evaluation of Modification to Computer Systems

As required by regulations, all maintenance work must be performed after the evaluation and approval of the work and must be consistent with the selected SLC methodology. Maintenance to a software system includes, among other things, the following:

Perfective maintenance or correcting the system because of new requirements and/or functions.
Adaptive maintenance or correcting the system because of a new environment, which could include new hardware, new sensors or controlled devices, new operating systems, new regulations.
Corrective maintenance or correcting the system because of detection of errors in the design, logic, or programming of the system. It is essential to recognize that the longer a defect is present in the software before it is detected the more expensive it is to fix it.
Preventive maintenance or correcting the system to improve future maintainability or reliability in order to provide a better basis for future enhancements.

Change management procedural control is in place when the validated system is released for use. The change management procedural control provides for the following activities:

Identifying and specifying the change.
Assessing risk, criticality, and impact of change.
Specifying testing requirements and acceptance criteria.
Implementing change after authorization.
Performing regression testing.
Reviewing of change(s) with an independent reviewer.
Updating system and user documentation to reflect implemented change(s).
Establishing of provisions for the management of ''emergency change,'' including expeditious documentation modifications.

SYSTEM DEVELOPMENT FILES

One key element to support the SLC is the availability and maintenance of system development files. The developer shall document the development of each system unit, system component, and configuration items in software development files.

The developer should establish a separate system development file for each unit or a logically related group of units. The developer should document and implement procedures for establishing and maintaining system development files. The developer should maintain the system development files until the retirement of the system. The system development files should be available to the agency review upon request. System development files may be generated, maintained, and controlled by automated means. To reduce duplication, system development files should not contain information provided in other documents or system development files. The set of system

Table 1 Period/events computer systems operational life

Period/event	Representative characteristics	Key practices
Early operational life	Phased roll outs	Problem reporting and maintenance
Maturity	Corrective, adaptive, perfective, and preventive maintenance	Operational audit Performance evaluation
Aging	Maintainability issues of obsolete technology (e.g., access to electronic records)	Periodic review Re-engineering analysis

Marcel Dekker, Inc.
270 Madison Avenue, New York, New York 10016

development files shall include (directly or by reference) the following information:

Design considerations and constraints.
Design documentation and data.
Schedule and status information.
Test requirements and responsibilities.
Verification and test procedures, and results.

CONCLUSION

The validation of computer systems in the U.S. FDA–regulated environment is an ongoing process that is integrated with the entire System Life Cycle (SLC). Quality to a software system is introduced by following the system life cycle and following the key validation elements.

REFERENCES

1. Food and Drug Administration. *Draft ''Guidance for Industry: 21 CFR Part 11; Electronic Records; Electronic Signatures Glossary of Terms''*; U.S. Department of Health and Human Services: Washington, DC, August, 2001.
2. Note that Part 820 requires that ''results'' of acceptance activities be recorded but not necessarily all raw data. ''Results'' must have audit trails. Be sensitive to the need for raw data during failure investigations under Corrective and Preventive Actions. Refer to Part 820 preamble, pp. 52631 and 52646.
3. FDA. *Guidance for the Industry: Computerized Systems Used in Clinical Trials*; April, 1999.
4. ANSI/IEEE Std 610.12-1990. *Standard Glossary of Software Engineering Terminology*; Institute of Electrical and Electronic Engineers: New York, 1990.
5. Herr, R.R.; Wyrick, M.L. *A Globally Harmonized Glossary of Terms for Communicating Computer Validation Key Practices*; PDA Journal of Pharmaceutical Science and Technology, March/April, 1999.
6. Cho, C.-Q. *Quality Programming: Developing and Testing Software Using Statistical Quality Control*; Wiley: New York, 1987.
7. López, O. Applying design controls to software in the FDA-regulated environment. J. cGMP Compliance **July, 1997**, *1* (4).
8. Grigonis, G.J.; Subak, E.J.; Wyrick, M.L. Validation key practices for computer systems used in regulated operations. Pharm. Technol. **June, 1997**.

BIBLIOGRAPHY

Annex 11 of the EU-GMP.

ANSI/AAMI SW68. *Medical Device Software—Software Life Cycle Processes*; Association for the Advancement of Medical Instrumentation, 2001.

ANSI/ISO/ASQ Q9000-3. *Quality Management And Quality Assurance Standards—Part 3: Guidelines for the Application of ANSI/ISO/ASQC Q90001-1994 to the Development, Supply, Installation and Maintenance of Computer Software*; 1997.

FDA Compliance Policy Guides for Computerized Drug Processing. *CGMP Applicability to Hardware and Software (CPG 7132a.11); Vendor Responsibility (CPG 7132a.12); Source Code for Process Control Application Programs (CPG 7132a.15); Input/Output Checking (CPG 7132a.07); Identification of ''Persons'' on Batch Production and Control Records (CPG 7132a.08); Enforcement Policy: 21 CFR Part 11; Electronic Records; Electronic Signatures (CPG 7153.17).*

Grigonis, Subak, Wyrick. Validation key practices for computer systems used in regulated operations. Pharm. Technol. **June, 1997**.

Good Automated Manufacturing Practice Forum (Rev 4). *Supplier Guide for Validation of Automated Systems in Pharmaceutical Manufacture*; ISPE, 2001.

Herr; Wyrick. A globally harmonized glossary of terms for communicating computer validation key practices. PDA J. Pharm. Sci. Technol. **March/April, 1999**.

IEC 62A/62304/Ed.1 (IEC 62A/338/NP). *Medical Device Software—Software Life Cycle Processes*; IEC, 2001.

ISO/IEC 1207. *Information Technology—Software Life Cycle Processes*; ISO/IEC, 1995.

ISO/IEC 12119. *Information Technology—Software Packages—Quality Requirements and Testing*; ISO/IEC, 1994.

López, O. FDA regulations of computer systems in drug manufacturing—13 years later. Pharm. Eng. **May/June, 2001**, *21* (3).

López, O. Implementing software applications compliant with 21 CFR part 11. Pharm. Technol. **March, 2000**.

López, O. *21 CFR Part 11 A Complete Guide to International Compliance*; Sue Horwood Publishing Limited, 2002, (http://www.computer-systems-validation.com), ISBN 09540706 7 4.

PIC/S Guidance. *Best Practices for Computerized Systems in Regulated 'GxP' Environments*; PIC/S, January, 2000.

Wyn, S. Regulatory requirements for computerized systems in pharmaceutical manufacture. Softw. Eng. J. **1996**, *11* (1).